

Unidad 2- Primeros pasos en PHP

Preparando el entorno

Para preparar un entorno de ejecución de PHP utilizaremos XAMPP que ya nos incluye la base de datos y el servidor web preparado con PHP.

Para instalarlo debemos realizar las siguientes acciones:

1. Descargar XAMPP: Entra a <https://www.apachefriends.org/index.html> y descarga la versión de XAMPP según el sistema operativo que tengas (Windows, macOS, Linux)
2. Instalar XAMPP: Una vez descargado, dale doble click al instalador de XAMPP y sigui las instrucciones en pantalla para completar la instalación (siguiente, siguiente, siguiente). Se recomienda instalarlo en la raíz del sistema para mayor acceso a los servicios (pero no mayor seguridad)
3. Iniciar los servicios: Después de instalar XAMPP, inicia XAMPP desde el ejecutable. Cuando abra, podemos iniciar los servicios de Apache y MySQL haciendo clic en los botones "Start" al lado de ellos en la pantalla de XAMPP.
4. Verificar la instalación: Abre tu navegador preferido y accede a <http://localhost> para verificar que XAMPP está funcionando. Deberías ver la página de inicio de XAMPP
5. Crear y ejecutar archivos PHP: Para ejecutar archivos PHP, crea tus archivos PHP en el directorio "htdocs" dentro del directorio de instalación de XAMPP. Por ejemplo, si instalaste XAMPP en "C:\xampp", poné tus archivos PHP en "C:\xampp\htdocs". Luego, accede a tus archivos PHP a través de "<http://localhost/nombreDelArchivo.php>"

Creación de un archivo PHP

Para crear un archivo PHP, debemos usar un editor de código (se recomienda utilizar Visual Studio Code <https://code.visualstudio.com/>). A continuación, deberemos crear un archivo con extensión PHP que contenga la referencia de script php

```
<?php  
  
echo "¡Hola, mundo!";  
  
?>
```

La apertura y cierre de un script php se realiza con “<?php” para la apertura y “?>” para el cierre (este último se puede omitir si solo se posee código php en el archivo ya que podemos embeber código php a un archivo con etiquetas html)

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Documento</title>
</head>
<body>
  <?php
    echo "Hola mundo";
    print("Hola mundo");
    printf("Hola mundo");
  ?>
</body>
</html>
```

Comentarios en PHP

La realización de comentarios en php es muy similar a otros lenguajes de programación.

```
<?php

    // Comentario de linea

    # Comentario de linea

    /*

    Comentario de bloque multi linea

    */
```

Tenemos dos tipos de comentarios de línea y un tipo de comentario de bloque.

Puedo usar las líneas de comentario en líneas vacías como el ejemplo anterior o como complemento en líneas de código

```
echo "Hola mundo"; //Muestra el mensaje Hola mundo
```

Variables en PHP

Las variables son espacios de almacenamiento de datos. Las variables poseen tres características: nombre, tipo de dato y valor (al ser un lenguaje de programación de tipado débil, podremos omitir la declaración de tipo de dato)

Para crear una variable (declararla y asignarle un valor) debemos realizar el siguiente código

```
<?php
    $x = 24;
    $y = "Christian";
```

Las condiciones para la creación de una variable son las de tener un nombre adecuado (el nombre de la variable comienza con el signo \$ para indicar el comienzo de una variable y es seguido por un carácter alfabético o guión bajo que luego podrá continuar con cualquier carácter alfanumérico), un valor (el mismo puede ser numérico, de texto, booleano, decimal, o de otros tipos primarios. Cuando se ingresa el dato también se le asigna a la variable el tipo de dato correspondiente)

Recuerda que las variables son case-sensitive así que las mayúsculas y minúsculas generan variables distintas cuando se utilicen para nombrar variables (la variable \$nombre es distinta a la variable \$Nombre) y que los nombres de variables que debe utilizar, deben ser afines a lo que se busca almacenar (no voy a almacenar)

Finalmente, si queremos ver de qué tipo es una variable podemos utilizar la función propia de PHP llamada var_dump()

```
<?php
```

```
    $edad = 24;
```

```
    $nombre = "Christian";
```

```
    var_dump($edad);
```

```
    var_dump($nombre);
```